SP-2153

1.00 0.50 19

A Necessary and Sufficient Condition for a Context-Free Grammar to be Unambiguous

Val Schorre

28 July 1965

SP-2153

SP a professional paper

A Necessary and Sufficient Condition for a Context-Free Grammar to be Unambiguous

Val Schorre

28 July 1965

SYSTEM DEVELOPMENT CORPORATION
2500 COLORADO AVE.
SANTA MONICA
CALIFORNIA

ABSTRACT

A condition is given for a context-free grammar to be unambiguous. This condition is proved to be both necessary and sufficient. A class of context-free grammers called first-character-recognition grammars (or fcr grammars) is defined. These grammars obviously satisfy the necessary and sufficient condition; consequently, they are unambiguous. It is shown to be a decidable question, whether a given grammar is an fcr grammar. Many programming languages can be described by fcr grammars; ALGOL can be so described, except for the distinction between arithmetic and Boolean expressions.

INTRODUCTION

This paper employs a formalism that has become standard for context-free languages. Ginsburg [1] has reported the results of a number of studies in this area. To provide background information, some results of those studies are summarized in the following paragraphs.

Writing two words together indicates concatenation; writing two sets of words together indicates complex product. For an arbitrary set E, E* is the set of all words over E, i.e., all the finite sequences of elements from E. In particular, E* contains the empty word ϵ. If w is a word, then $|w|$ is the number of elements in w. If xyz = u, then y is said to be a subword of u, and x is said to be an initial subword of u, and z is said to be a terminal subword of u. A subword of u that is distinct from u is called a proper subword.

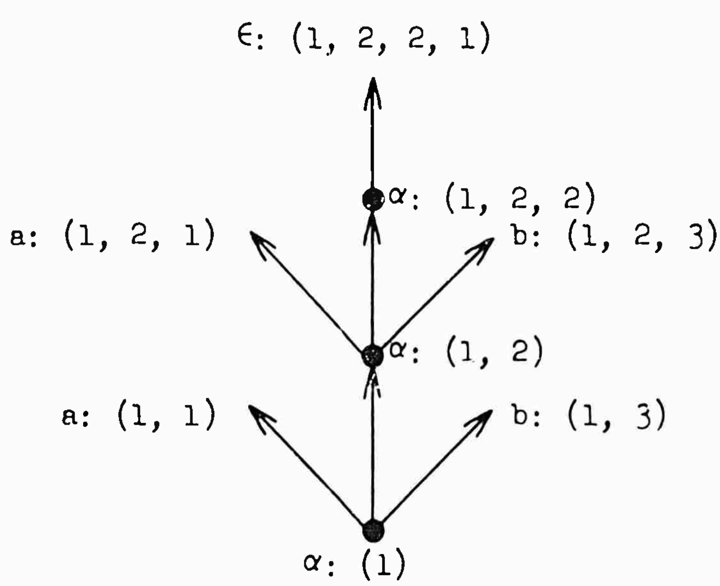
Definition. A grammar G is a 4-tuple (V, Σ, P, σ), where V is a finite set ("vocabulary"), Σ is a subset of V ("Letters"), σ is an element of $V-\Sigma$, and P (the set of "production") is a finite set of ordered pairs of the form $\xi \rightarrow w$, with ξ in $V-\Sigma$ and w in V*.

Definition. By a <u>node</u> from v in V is meant a sequence of positive integers $(1, i_1, i_2, i_3, \ldots, i_k)$ with $0 \leq k$ such that if $1 \leq k$ there is a sequence of productions $v_1 \rightarrow w_1, v_2 \rightarrow w_2, v_3 \rightarrow w_3, \ldots, v_k \rightarrow w_k$ such that: (1) v_1 is v, (2) v_{j+1} is the i_j-th term of w_j, for $j < k$, (3) if $w_k = \epsilon$, $i_k = 1$ and (4) if $w_k \neq \epsilon$, $i_k \leq |w_k|$.

When $1 \leq k$ and $w_k \neq \epsilon$, we call the i_k-th term of w_k the <u>label</u> of the node $(1, i_1, i_2, i_3, \ldots, i_k)$. When $1 \leq k$ and $w_k = \epsilon$, we call ϵ the <u>label</u> of the node. The letter v is the label of the node (1).

The node $(1, i_1, i_2, i_3, \ldots, i_k)$ is called <u>terminal</u> if $w_k = \epsilon$ or the i_k-th term of w_k is in Σ. The sequence (1) is considered a terminal node from all v in Σ.

Definition. The statement that T is a <u>generation tree</u> from v in V means that T is a collection of nodes from v such that: (1) the sequence (1) is in T; (2) if $(1, i_1, i_2, i_3, \ldots, i_k)$ is a non-terminal node of T, then there is one and only one j such that $(1, i_1, i_2, i_3, \ldots, i_k, j)$ is in T: and (3) if $(1, i_1, i_2, i_3, \ldots, i_k)$ is in T then $(1, i_1, i_2, i_3, \ldots, i_{k-1})$ is also in T. The sequence (1) is called the <u>root</u> of the tree and its label is v. From this definition it follows that the generation tree for v in Σ contains only the node (1).



Generation Tree for the Word aabb in a Grammar Whose Productions are $\alpha \to a\, \alpha\, b$ and $\alpha \to \epsilon$.

Definition. By the length of the generation tree T is meant one less than the maximum length of the sequences of integers in T. The length of the generation tree in the example is 3.

Notation. If each of $(1, i_1, i_2, i_3, \ldots, i_m)$ and $(1, j_1, j_2, j_3, \ldots, j_n)$ is a node from v then write $(1, i_1, \ldots, i_m) < (1, j_1, \ldots, j_n)$ to mean that either (1) $m < n$ and $i_k = j_k$ for $1 \leq k \leq m$ or (2) if k is the least integer such that $i_k \neq j_k$ then $i_k < j_k$.

Definition. Let all the terminal nodes of the generation tree T be arranged in a sequence $N_1, \ldots, N_k$ such that $N_i < N_{i+1}$, for $1 \leq i \leq k$. Let B_i be the label of N_i, for $1 \leq i \leq k$. The word $B_1 \ldots B_k$ is called the <u>sentence</u> of T and is denoted by S(T).

Definition. For v in V, $L(v) = \{ x \mid \exists$ a generation tree from v whose sentence is $x\}$. L(v) is called the <u>language</u> of v. For v_1, v_2 in V, let $L(v_1 v_2) = L(v_1) L(v_2)$. The function L is now defined for all v in $V \cup V^2$. The language of a grammar is usually defined in terms of a sequence of steps in which members of V* are rewritten, see Bar-Hillel [2]. It should be clear to the reader that these two definitions are equivalent.

Definition. The grammar $G = (V, \Sigma, P, \sigma)$ is said to be <u>ambiguous</u> at v in V if there are two distinct generation trees from v which produce the same word. The entire grammar is said to be ambiguous, if it is ambiguous at σ.

Definition. A grammar is said to be binary if each production is either of the form $v \to \epsilon$ or $v \to v_1 v_2$ with v_1, v_2 in V.

THE NECESSARY AND SUFFICIENT CONDITION

Definition. The set of terminal words that will be associated with x in $V \cup V^2$ is defined by $H(x) = \{ z | \exists$ u, v both in $L(x)$ such that $u z = v\}$.

Definition. The set of initial words that will be associated with x in $V \cup V^2$ is defined by $K(x) = \{ z | \exists$ u, v both in $L(v)$ such that $z u = v\}$.

Theorem. It is a necessary and sufficient condition for a binary grammar $G = (V, \Sigma, P, \sigma)$ to be unambiguous at all v in V that: (1) if $\xi \to x_1 x_2$ and $\xi \to y_1 y_2$ are two distinct productions of G then $L(x_1 x_2) \cap L(y_1 y_2) = \emptyset$ and (2) if $\xi \to x y$ is a production of G then $H(x) \cap K(y) = \emptyset$.

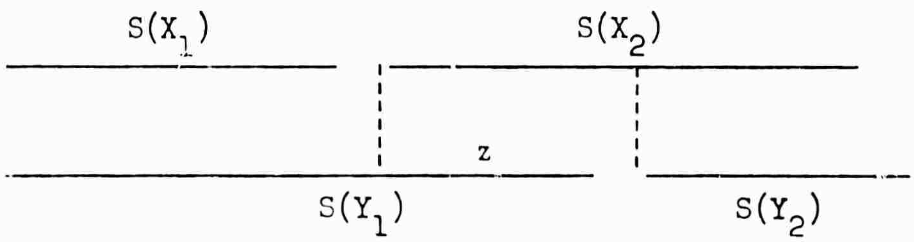
Proof. Since the condition is obviously necessary, we proceed to prove that it is sufficient. The grammar G will now be shown unambiguous by induction on the lengths of the generation trees. Assume that for all ξ in V there do not exist two distinct generation trees of length $\leq n$ from ξ which produce the same word in $\Sigma*$. Assume furthermore that there does exist a v in V and a pair of generation trees X and Y from v with length $\leq n+1$ which produce the same word in $\Sigma*$.

Case 1. The tree X begins with the production $v \to x$ and the tree Y begins with a different production $v \to y$, where x and y are in $V^2 + \epsilon$. Then $S(X) = S(Y)$ is in $L(x) \cap L(y)$, which is contrary to the hypothesis.

Case 2. Both trees begin with the same production $v \rightarrow u_1 u_2$. Just below the root of X are two subtrees X_1 and X_2 from u_1 and u_2 respectively. Let Y_1 and Y_2 be similar subtrees of Y.

Case 2.1 $S(X_1) = S(Y_1)$. Then $S(X_2) = S(Y_2)$. Since the subtrees X_1 and Y_1 have length $\leq n$, they cannot be involved in an ambiguity from u_1, therefore $X_1 = Y_1$. Similarly, $X_2 = Y_2$ and thus X = Y which contradicts the contrary assumption.

Case 2.2 $S(X_1)$ is a proper initial subword of $S(Y_1)$. Let z be a word such that $S(X_1)\ z = S(Y_1)$.



Since z is in $H(u_1)$ and $K(u_2)$, the hypothesis of the theorem is contradicted.

Case 2.3 $S(Y_1)$ is a proper initial subword of $S(X_1)$. Similar to case 2.2.

Q.E.D.

FIRST-CHARACTER-RECOGNITION GRAMMARS

Definition. Let F(x) denote the set of all first letters of words in L(x) and let E(x) denote the set of all first letters of words in H(x). Let Q(x) be the predicate "ϵ is in L(x)."

Definition. The statement that $G = (V, \Sigma, P, \sigma)$ is a <u>first-character-recognition grammar</u> (or an fcr grammar), G is a binary grammar and means that (1) if $\xi \rightarrow x_1\ x_2$ and $\xi \rightarrow y_1\ y_2$ are two distinct productions of G, then $F\ (x_1\ x_2) \cap F\ (y_1\ y_2) = \emptyset$ and either $\sim Q\ (x_1\ x_2)$ or $\sim Q\ (y_1\ y_2)$ and (2) if $\xi \rightarrow x\ y$ is a production of G, then $E(x) \cap F(y) = \emptyset$.

Theorem. Every fcr grammar is unambiguous.

This theorem obviously follows from the necessary and sufficient condition. The computability of the predicate $Q(x)$ follows from the theorem, that it is decidable whether any word, including ϵ, belongs to a language. For a proof of this theorem see Chomsky [4] or Bar-Hillel [2]. To prove that $F(x)$ is computable we can make use of the theorem of Ginsburg [3] that states that the image of a context-free language under a finite state transducer is itself a context-free language. The transducer needed is a simple one that outputs the first letter of each word given it. From the given grammar, the transducer theorem gives us a new grammar whose language is $F(x)$. We use the decidability algorithm mentioned above to determine which letters are words of the new grammar.

At this point, more direct methods of computing $Q(x)$ and $F(x)$ will be given. These methods employ ascending sequences, similar to those used by Bar-Hillel [2].

For v, v_1, v_2 in V and $1 \leq n$

$$Q_1\ (v) = v \rightarrow \epsilon$$

$$Q_{n+1}\ (v) = Q_n\ (v) \text{ or } \exists\ u_1, u_2\ (v \rightarrow u_1\ u_2 \text{ and } Q_n\ (u_1) \text{ and } Q_n\ (u_2))$$

$$Q_n\ (v_1\ v_2) = Q_n\ (v_1) \text{ and } Q_n\ (v_2)$$

It can be easily proven that there exists k such that for p in $V \cup V^2$

$Q_1 (p) \Rightarrow Q_2 (p) \Rightarrow Q_3 (p) \Rightarrow \dots \Rightarrow Q_k (p) = Q_{k+1} (p) = \dots$ and $Q_k (p) = Q (p)$

For v, v_1, v_2 in $V - \Sigma$ and $1 \leq n$

$F_1 (v) = \{ x \mid x \text{ in } \Sigma \text{ and either } v = x \text{ or } \exists\, y\ (v \to x\ y)\}$

$F_{n+1} (v) = F_n(v) \cup \{ x \mid \exists\, u_1, u_2\ (v \to u_1\ u_2 \text{ and either } x \text{ in } F_n (u_1) \text{ or } Q_n (u_1)$
$\text{and } x \text{ in } F_n (u_2) \}$

$F_n (v_1\ v_2) = F_n (v_1) \cup \{ x \mid Q_n (v_1) \text{ and } x \text{ in } F_n (v_2) \}$

It can be easily proven that there exists k such that for p in $V \cup V^2$

$F_1 (p) \subseteq F_2 (p) \subseteq F_3 (p) \subseteq \dots \subseteq F_k (p) = F_{k+1} (p) = \dots$ and $F_k (p) = F (p)$

The function $E(p)$ is not so easy to handle. As before, a chain will be defined that converges to a function $E_k (p)$. Suppose we make a new condition by replacing $E (p)$ with $E_k (p)$ in the sufficient condition already given. These two conditions will be proven to be equivalent. The method of proof is to show that for all p in V, $E_k (p) \subseteq E (p)$ and if the new condition is satisfied, then for all p in V, $E (p) \subseteq E_k (p)$.

For v in V and $1 \leq n$

$E_1 (v) = \emptyset$

$E_{n+1} (v) = E_n (v) \cup \{ x \mid \exists\, u_1, u_2\ (v \to u_1\ u_2 \text{ and either } x \text{ in } E_n (u_2) \text{ or}$
$Q_n (u_2) \text{ and } x \text{ in } E_n (u_1) \text{ or } Q_{n+1} (v) \text{ and } x \text{ in } F_n (u_1))\}$

It can easily be proven that there exists k such that for p in $V \cup V^2$

$$E_1 (p) \subseteq E_2 (p) \subseteq \dots \subseteq E_k (p) = E_{k+1} (p) = \dots$$

Lema. For all v in V, $E_k (v) \subseteq E (v)$.

This is also easily proven.

Definition. For p in V and $1 \leq n$, $E^n (p) = \{x \mid \exists$ two generation trees X and Y from p of length $\leq n$ and a word w in $\Sigma*$ such that $S (X) w = S (Y)$ and x is the first letter of w $\}$.

Lema. Suppose that $G = (V, \Sigma, P, \sigma)$ is a binary grammar such that (1) if $\xi \to x_1 x_2$ and $\xi \to y_1 y_2$ are two distinct productions of G then $F (x_1, x_2) \cap F (y_1, y_2) = \emptyset$ and either $\sim Q (x_1 x_2)$ or $\sim Q (y_1 y_2)$ and (2) if $\xi \to x y$ then $E_k (x) \cap F (y) = \emptyset$. For v in $V\text{-}\Sigma$ and $1 \leq n$, $E^n (v) \subseteq E_k (v)$.

Proof.

The theorem is proven by induction in the following manner. Let n be some integer ≥ 1. Assume that for all v, $E^n (v) \subseteq E_k (v)$. Let p be in $V\text{-}\Sigma$ and let s be in $E^{n+1} (p)$.

It must be shown that $E^{n+1} (p) \subseteq E_k (p)$, in other words, that s is in $E_k (p)$.

There exists a pair of generation trees X and Y from p such that (1) both X and Y have length $\leq n+1$ and (2) $\exists$ z (S(X) z = S (Y) and s is the first letter in z).

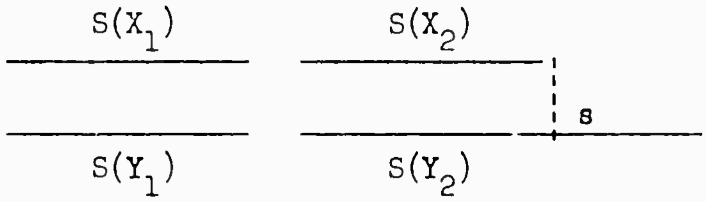
Case 1 $S(X) = \epsilon$

Let $p \to y_1 y_2$ be the first production of the tree Y. The length of the subtree from y_1 is $\leq n$; therefore, s is in $F_n (y_1)$. Since $Q_{n+1} (p)$ is true, it follows that s is in $E_{n+1} (p)$ and, consequently, in $E_k (p)$.

Case 2 $S(X) \neq \epsilon$

Let $p \to x$ and $p \to y$ be the first productions of the trees X and Y respectively, with x and y in V^2. Assume these productions are distinct. Then $F(x) \cap F(y) = \emptyset$. This is impossible since S(X) is an initial subword of S(Y). Let $p \to v_1 v_2$ be the first production of both trees X and Y. Just below the root of X are two subtrees X_1 and X_2 from v_1 and v_2 respectively. Let Y_1 and Y_2 be similar subtrees for Y.

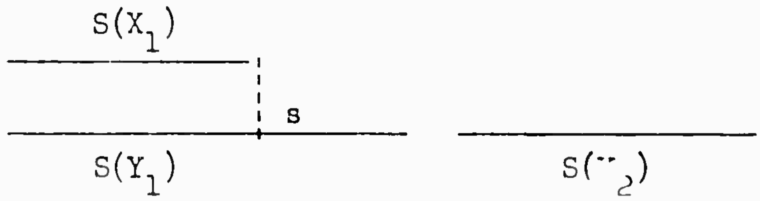
Case 2.1 $S(X_1) = S(Y_1)$



s is in $E^n(v_2)$; s is in $E_k(v_2)$; s is in $E_{k+1}(p)$; s is in $E_k(p)$

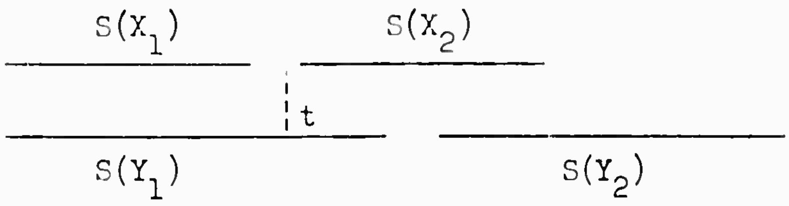
Case 2.2 $S(X_1)$ is a proper initial subword of $S(Y_1)$

Case 2.2.1 $S(X_2) = \epsilon$



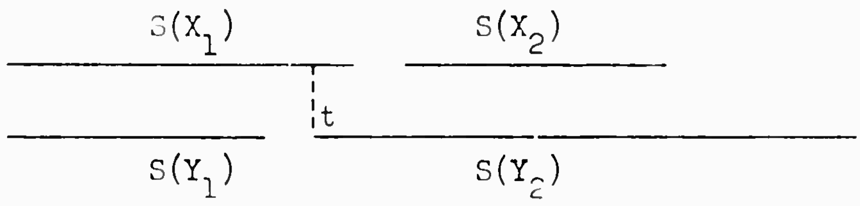
s is in $E^n(v_1)$; s is in $E_k(v_1)$; $Q_k(v_2)$ is true, s is in $E_{k+1}(p)$; s is in $E_k(p)$.

Case 2.2.2 $S(X_2) \neq \epsilon$



Let t be the first letter in $S(X_2)$. t is in $E^n(v_1)$; t is in $E_k(v_1)$; t is in $F(v_2)$; $E_k(v_1) \cap F(v_2) \neq \emptyset$; Contradiction

Case 2.3 $S(Y_1)$ is a proper initial subword of S(X.)



Similar to case 2.2.2

Q.E.D.

Theorem. If $G = (V, \Sigma, P, \sigma)$ is a binary grammar, then G is an fcr grammar if and only if (1) if $\xi \rightarrow x_1 x_2$ and $\xi \rightarrow y_1 y_2$ are distinct productions then $F(x_1 x_2) \cap F(y_1 y_2) = \emptyset$ and either $\sim Q(x_1 x_2)$ or $\sim Q(y_1 y_2)$ and (2) if $\xi \rightarrow x\ y$ then $E_k(x) \cap F(y) = \emptyset$.

Proof

Part 1. If statement 1 then statement 2.

$E_k(x) \subseteq E(x)$; $E(x) \cap F(y) = \emptyset$; $E_k(x) \cap F(y) = \emptyset$.

Part 2. If statement 2 then statement 1.

For v in $V-\Sigma$ and $1 \leq n$, $E^n(v) \subseteq E_k(v)$.

For v in $V-\Sigma$, $E(v) \subseteq E_k(v)$.

For v in Σ, $E(v) = \emptyset$.

For v in V, $E(v) \subseteq E_k(v)$; $E_k(v) \subseteq E(v)$; $E(v) = E_k(v)$.

Q.E.D.

CONCLUSION

Grammars that are used in a top-to-bottom syntax scan without backup are almost always fcr grammars. Such grammars have been used in syntax-directed compilers by Schorre [6], Schmidt [5], and Schneider and Johnson [7]. The only difficulty in writing such a grammar for ALGOL appears to be that the distinction between algebraic and Boolean expressions is lost because they both begin with an arbitrary number of open parentheses. In other words, the language of the rewritten grammar would contain all the words in ALGOL and, in addition, words that had algebraic expression where only Boolean expressions should be (for example, ... IF A + B THEN ...).

CITED REFERENCES

(1) Ginsburg, Seymour. "A survey of ALGOL-like and context free language theory," invited pape. at the IFIP Working Conference on "Formal Language Description Languages," in Austria, 1964; SDC TM-738/006/00.

(2) Bar-Hillel, Y., Perles, M., and Shamir, E. "On formal properties of simple phrase structure grammars," Zeitschrift fur Phonetik, Sprachwissenschaft und Kommunikationsforschung, Vol. 14, 1961, pp. 143-172.

(3) Ginsburg, S. and Rose, G. F. "Operations which preserve definability in languages," Journal of the Association for Computing Machinery, Vol. 10, 1963, pp. 175-195.

(4) Chomsky, N. "On certain formal properties of grammars," Information and Control 2 (1959) 137-167.

(5) Schmidt, L., "Implementation of a symbol manipulator for heuristic translation," 1963 ACM National Conference, Denver, Colorado.

(6) Schorre, D. V., "META II, a syntax-oriented compiler writing language," 1964 ACM National Conference, Philadelphia.

(7) Schneider, F. W. and Johnson, G. D., "A syntax-directed compiler-writing compiler to generate efficient code," 1964 ACM National Conference, Philadelphia.

UNCITED REFERENCES

Cantor, D. "On the ambiguity problem of Backus systems," Journal of the Association for Computing Machinery, Vol. 9, 1962, pp. 477-479

Chomsky, N. and Schutzenberger, M. P. "The algebraic theory of context-free languages," Computer Programming and Formal Systems, edited by Braffort and Hirschberg, North-Holland Publishing Co., Amsterdam, 1963.

Floyd, Robert W. "On ambiguity in phrase structure languages, Communications of the ACM, Vol. 5, No. 10 (October 1962), page 526.

UNCLASSIFIED

DOCUMENT CONTROL DATA - R&D

1 ORIGINATING ACTIVITY (Corporate author) SYSTEM DEVELOPMENT CORPORATION Santa Monica, California	2a. 2b.

3 REPORT TITLE

A NECESSARY AND SUFFICIENT CONDITION FOR A CONTEXT-FREE GRAMMAR TO BE UNAMBIGUOUS

4 DESCRIPTIVE NOTES (Type of report and inclusive dates)

5 AUTHOR(S) (Last name, first name, initial)

Schorre, D.V.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
28 July 1965	15	10

8a. CONTRACT OR GRANT NO. Government Contracts	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	SP-2153
c	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d	

10. AVAILABILITY/LIMITATION NOTICES This document has been cleared for open publication and may be disseminated by the Clearing House for Federal Scientific & Technical Information.

11. SUPPLEMENTARY NOTES	12. SPONSOR

13 ABSTRACT

A condition is given for a context-free grammar to be unambiguous. This condition is proved to be both necessary and sufficient. A class of context-free grammers called first-character-recognition grammars (or fcr grammars) is defined. These grammars obviously satisfy the necessary and sufficient condition; consequently, they are unambiguous. It is shown to be a decidable question, whether a given grammar is an fcr grammar. Many programming languages can be described by fcr grammars; ALGOL can be so described, except for the distinction between arithmetic and Boolean expressions.

Unclassified

A-2735

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Context-Free Unambiguous Grammar						